

ARE HEISENBERG

2016



Moro MANE, Fabien TANG, Marie-Céline BESHARA, Nicolas MILON

Sommaire

Introduction	3
Modélisation.....	3
Génération de la carte	3
Génération des personnages.....	3
Interactions entre personnages	3
Analyse de la modélisation.....	3
Réalisation	4
Fonction Albuquerque	4
Fonction Ville	4
Fonction Junkies.....	4
Fonction Stade_final	5
Fonction Pourcentage_de_junkie.....	5
Fonction Analyse.....	5
Résultats Analyse	6
Résultat 1	6
Résultat 2	6
Résultat 3	6
Conclusion.....	6
Amélioration du modèle	7
Conclusion.....	7
Conclusion	7

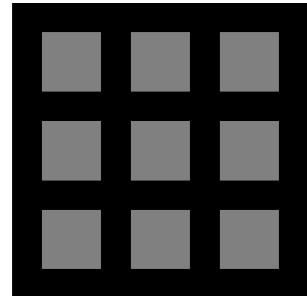
Introduction

Suite aux quatre semaines de simulation de feu de forêt en python nous devons créer notre propre programme de simulation. Nous avons tout d'abord pensé à des sujets comme des invasions de zombies ou bien de grippe mais, après mûre réflexion (et le fait que ses thèmes intéressaient également d'autre groupe), notre choix s'est finalement porté sur le trafic de drogue d'où le titre de notre projet : « ARE - Heisenberg » tiré du fameux Walter H. White de la série Breaking Bad.

Modélisation

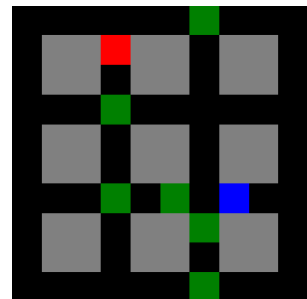
Génération de la carte

Pour réaliser notre projet nous avons codé une fonction générant un quadrillage représentant une ville américaine avec les cases noires représentant les routes et les cases grises les immeubles.



Génération des personnages

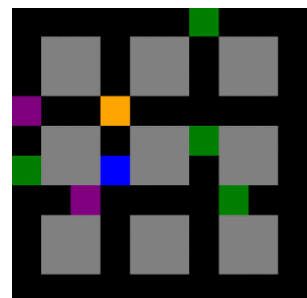
Puis nous avons codé une seconde fonction permettant de générer, à partir des pourcentages donnés par l'utilisateur, une population « saine » (cases vertes), des dealers (cases rouges) et des policiers (cases bleues).



Interactions entre personnages

Le principe de notre simulation est celui-ci :

- La population et les policiers se déplacent pas par pas de façon aléatoire dans la ville.
- Si une personne saine rencontre un dealer elle devient un junkie (case violette).
- Si un policier rencontre un dealer il l'arrête (la case du dealer devient jaune puis disparaît au tour suivant).
- La simulation s'arrête après l'arrestation du dernier dealer, mais la position du dernier dealer reste afficher en jaune.



Analyse de la modélisation

Enfin, afin d'avoir des résultats concrets, nous avons codé plusieurs fonctions permettant:

- d'obtenir l'état final d'une ville
- de compter le nombre de junkie sur le nombre total de la population (hors policier et dealer)
- de réaliser n-fois la simulation et en sortir une moyenne, un écart-type et une variance
- de tracer le graphique d'une moyenne, d'une variance ou d'un écart-type.

Réalisation

Fonction Albuquerque

La fonction permet de générer les routes (en noir) et les buildings (en gris) à l'aide de boucle for. Notre fonction permet aussi de définir la taille de notre ville et la taille d'un immeuble.

Fonction Ville

La fonction ville permet dans un premier temps de générer une ville (Albuquerque), puis de choisir le pourcentage de dealer que l'on souhaite placer dans celle-ci. Parmi les coordonnées des routes d'Albuquerque placées au préalable dans une liste, on tire aléatoirement un nombre de coordonnées dans cette liste afin d'atteindre le pourcentage indiqué par l'utilisateur puis on place les dealers aux coordonnées tirés.

Exemple : S'il y a 100 cases disponibles représentant les routes d'Albuquerque et que l'utilisateur veut 60% de junkies, on récupéra aléatoirement dans la liste 60 cases de route. On fait de même pour la population (cases vertes) et les policiers (cases bleues).

Fonction Junkies

Interactions entre dealers, policiers et population

Avant tout déplacement, la fonction "junkies" s'occupe d'appliquer les interactions décrites plus haut (cf. Modélisation, Interactions entre personnages).

Deux boucles "for" imbriquées permettent de parcourir toutes les cases de la ville, afin de trouver les dealers. Une fois les dealers localisés, on commence par tester si un policier se trouve sur une case adjacente :

- S'il y en a un, le dealer est arrêté et sa case change de valeur (Il est représenté en jaune dans l'affichage graphique). Au prochain tour, il disparaîtra, sa case changera de valeur pour redevenir une route.
- S'il n'y a pas de policiers, on regarde si une personne saine se trouve près du dealer. Si c'est le cas, la case représentant cette personne change de valeur afin de devenir un junkie qui ne commencera à bouger qu'au prochain tour.

Déplacement de la population saine et des junkies

On parcourt d'abord toutes les cases de la matrice pour trouver une personne saine. Une fois une personne saine trouvée, on regarde quels sont ses déplacements possibles puis on les ajoutent dans une liste.

Le déplacement vers une case est impossible :

- si la case est occupée par une autre personne,
- si la case est un bâtiment,
- si la case est en dehors de la matrice.

On tire aléatoirement une position dans cette liste et on change la valeur de la case tirée pour montrer qu'une personne saine se trouve désormais à cette case et son ancienne position devient une route. Le déplacement des junkies et des policiers suit exactement le même principe.

Déplacement des policiers

Le déplacement des policiers suit le même principe que les déplacements de la population mais lorsqu'un policier se trouve sur la même ligne ou colonne qu'un dealer, il est attiré vers celui-ci. Si un policier se trouve à côté d'un dealer, il l'arrête.

Afin d'exploiter notre modélisation et d'obtenir des résultats concrets, nous avons codé plusieurs petites fonctions :

Fonction Stade final

Une première fonction `stade_final` permet d'obtenir à l'aide d'une boucle `while` l'état final d'une ville en appliquant la fonction `Junkies` à la ville tant qu'il y a un dealer.

Fonction Pourcentage de junkie

Cette fonction permet d'obtenir le nombre de junkie sur la population totale (hors policier et dealer) en parcourant la ville à l'aide de deux boucles `for`, des conditions et des compteurs.

Fonction Analyse

La fonction prend en paramètre un pourcentage de dealers, de policiers, de populations, un nombre `n` correspondant au nombre de création d'une nouvelle ville avec les pourcentages indiqués. À l'aide d'une boucle `while`, des fonctions `pourcentage de junkie` et `stade final`, cette fonction permet d'obtenir l'écart-type, la variance et la moyenne du pourcentage de junkies qu'il y a à l'état final des `n`-ville.

Résultats Analyse

Grâce à la fonction Analyse, nous avons obtenu plusieurs graphiques avec des valeurs plus ou moins étalées. Ci-dessous les résultats de nos analyses en modifiant les paramètres.

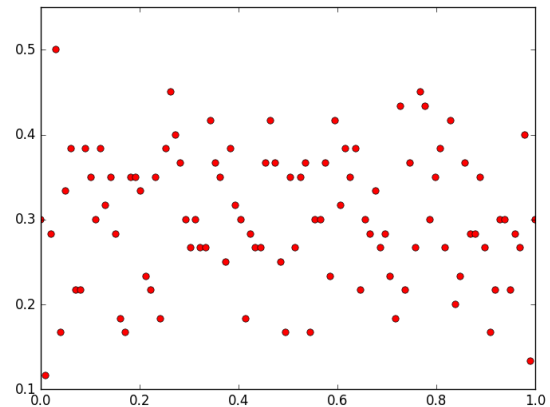
Résultat 1

20% Population

1% Dealers

1% Policiers

→ Moyenne de junkies entre 10% et 50%



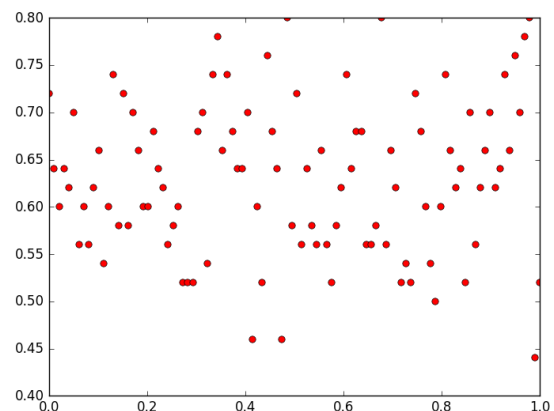
Résultat 2

20% Population

10% Dealers

1% Policiers

→ Moyenne de junkies entre 50% et 80%



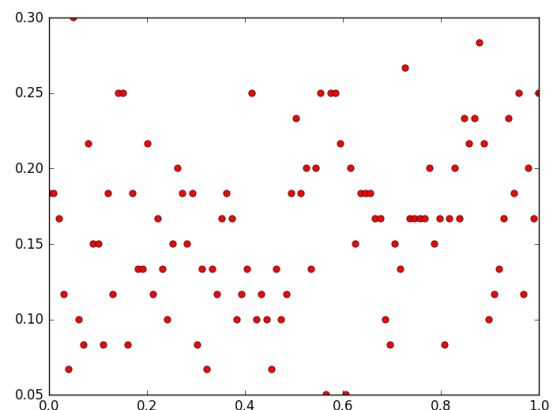
Résultat 3

20% Population

1% Dealers

10% Policiers

→ Moyenne de junkies entre 5% et 30%



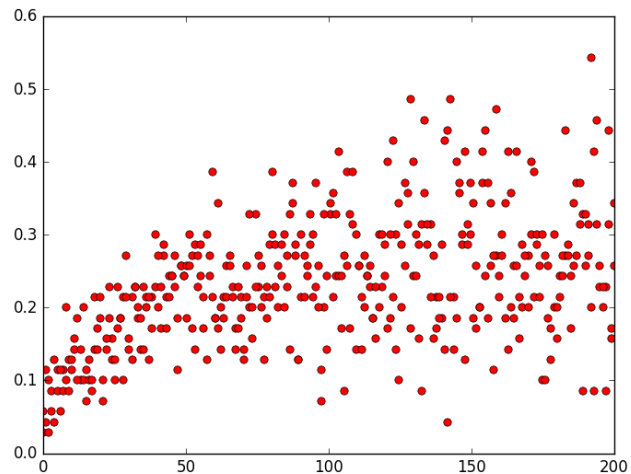
Conclusion

- Plus il y a de policiers, moins il y a de junkies,
- Plus il y a de dealers, plus il y a de junkies,
- L'étalement correspond aux caractères aléatoires des déplacements des policiers, des junkies et de la population.

Amélioration du modèle

Afin d'avoir des résultats plus réalistes, nous avons mis en place une probabilité de devenir Junkie. Lorsqu'une personne saine se trouve à côté d'un dealer, elle a cette fois-ci 1 chance sur 2 de devenir un junkie.

Nous avons réalisé une seconde fonction Analyse (Analyse_2) permettant cette fois-ci d'obtenir l'évolution du nombre de junkie dans les villes en fonction du nombre de pas réalisé.



50% populations, 1% Dealers, 1% Policiers

Conclusion

- Plus le nombre de pas augmente, plus il y a de Junkies,
- Plus le nombre de pas augmente, plus l'étalement est important car les déplacements sont aléatoire.

Conclusion

En conclusion, notre simulation et les résultats de nos analyses ont permis de mettre en avant l'évolution de la dépendance d'une ville en fonction d'un pourcentage de dealers et de policiers. Grâce à nos résultats, nous en avons conclu :

- Plus il y a de policiers, moins il y a de junkies,
- Plus il y a de dealers, plus il y a de junkies,
- L'étalement correspond aux caractères aléatoires des déplacements des policiers, des junkies et de la population.
- Plus le nombre de pas augmente, plus il y a de Junkies,
- Plus le nombre de pas augmente, plus l'étalement est important car les déplacements sont aléatoire.

Blog: <https://areheisenberg.wordpress.com/>